ELSEVIER

Journal of Computational and Applied Mathematics 82 (1997) 79–91

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Integration schemes for highly oscillatory DAEs with applications to circuit simulation

G. Denk[a,*], Chr. Penski[b]

[a] *Siemens AG, Corporate Research & Development, D-81390 München, Germany*

[b] *Technische Universität München, Mathematisches Institut/FORTWIHR, D-80290 München, Germany*

Received 27 August 1996; received in revised form 23 December 1996

Abstract

A new numerical integration scheme for the simulation of differential–algebraic equations is presented. In the context of the computer-aided design of electronic circuits, the modeling of highly oscillatory circuits leads to oscillatory differential–algebraic equations. Standard schemes can solve these equations neither efficiently nor reliably. To overcome the problems of classical numerical methods, the new discretization scheme is based on the *principle of coherence* due to Hersch in combination with a multistep approach. A Fortran77 implementation of the presented integration scheme reduces the simulation time for a quartz-controlled oscillator to about 2% compared with standard methods. Therefore, it is a useful tool for the design of highly oscillatory circuits.

Keywords: Differential–algebraic equation; Index; Circuit simulation; Weierstrass–Kronecker canonical form; Oscillatory solutions; Multistep method

AMS classification: 65L05; 65L06

1. Introduction

Circuit simulation is a standard task for the computer-aided design of electronic circuits. From a mathematical point of view, an electronic circuit consists of the characteristic element equations for the devices (resistor, MOSFET, etc.) and the device parameters. By means of the modified nodal analysis (MNA) [10] these element equations are directly inserted into Kirchhoff's current law which is applied for every node except ground. Additionally, the equations for the branch currents which have no simple admittance form have to be considered. This yields the following system of *differential–algebraic equations* (DAEs):

$$C\dot{x}(t) + Bx(t) = f(t,x(t)), \tag{1}$$

* Corresponding author. E-mail: Georg.Denk@mchp.siemens.de.

PII S0377-0427(97)00042-3

where $x : \mathbb{R} \to \mathbb{R}^n$ denotes the vector of the nodal voltages, possibly extended by the branch currents through voltage-defining elements. $C \in \mathbb{R}^{n\times n}$ represents the 'capacitance' matrix (usually singular), $B \in \mathbb{R}^{n\times n}$ the 'conductance' matrix and $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ assembles the nonlinear parts of the circuit and the independent sources.

It is important to note that the matrices C and B and the right-hand side $f(t,x)$ depend only on the circuit under investigation and the models used, they do not arise from a linearization of a nonlinear DAE. In a computer-aided design approach for circuit simulation, it is not possible to change this decoupling of the DAE. Therefore, we assume in the following given matrices C and B which can not be altered.

From a mathematical point of view, the transient analysis – i.e., the solution of Eq. (1) – is well-understood for most circuits. For highly oscillatory circuits, however, solving the DAEs with standard integration schemes (like DASSL [2]) raises several problems. In order to get a reliable approximation for the exact solution of the DAE, the standard schemes have to use very small step sizes in comparison with the integration interval. This holds even for $f(t,x(t)) \equiv 0$, as the waveform of the solution has to be approximated by polynomials and this requires at least some integrations steps per period. The integration scheme presented here allows for this special case unlimited step sizes and is able to jump over several periods without loss of accuracy. Similar properties hold for 'small' $f(t,x(t))$.

Besides this inefficiency, standard methods are not reliable for the integration of highly oscillatory systems: as the error control is based on polynomials, it can not adequately approximate the trigonometric characteristics of the oscillations.

Many authors have developed efficient numerical integration schemes for oscillatory ordinary differential equations (ODEs). Most of the methods deal with the special second order ODE $x''(t) = f(t,x(t))$, cf. [1, 9, 12]. However, there are no established methods for oscillatory first order ODEs. This becomes even worse, if oscillatory DAEs have to be solved.

To cope with this problem a new integration scheme is presented in this paper. In Section 2 the DAE (1) will be examined in an analytical way. We discuss the index concept arising by DAEs, the simultaneous matrix decomposition of the matrix pair (C,B), and we present the exact solution for a homogeneous DAE and for DAEs with polynomial right-hand sides $f(t,x(t))$. According to the principle of coherence stated at the end of Section 2, we will develop and investigate the efficient integration scheme MUNICH in Section 3. The last section shows the efficiency of MUNICH simulating a quartz-controlled oscillator for a PLL circuit which is a part of a Siemens ISDN Echo Cancel Chip.

2. Analysis of the DAE

The difficulties of the numerical treatment of DAEs can be characterized by the index of the DAE. The index concept can be defined in the following way [2]:

Definition 1. The equation

$$F(t,x(t),\dot{x}(t)) = 0,$$

with $x:\mathbb{R}\to\mathbb{R}^n$ and $F:\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^n$, has *differential index di*, if di is the minimal number of analytical derivatives

$$F(t,x,\dot{x})=0,\qquad \frac{\mathrm{d}F(t,x,\dot{x})}{\mathrm{d}t}=0,\ldots,\qquad \frac{\mathrm{d}^{di}F(t,x,\dot{x})}{\mathrm{d}t^{di}}=0, \tag{2}$$

such that (2) can be transformed into an explicit ordinary differential equation (ODE)

$$\dot{x}(t)=H(t,x(t)),$$

with $H:\mathbb{R}\times\mathbb{R}^n\to\mathbb{R}^n$, by algebraic manipulations.

For most of the circuit models the index is 1 or 2. In the following, we assume a unique solution of Eq. (1) which holds for all reasonable circuits.

For the construction of an efficient integration scheme, Eq. (1) has to be analyzed in detail. The next theorem (cf. [3]) is the base for the following investigations:

Theorem 2. *If the matrix pencil* $\{\lambda\cdot C+B,\lambda\in\mathbb{C}\}$ *is regular (i.e.,* $\det(\lambda C+B)\not\equiv 0$*), then the so-called WKCF-decomposition (Weierstrass–Kronecker canonical form)*

$$C=P^{-1}\left(\begin{array}{c|c} I_{n_d} & 0\\ \hline 0 & N\end{array}\right)Q^{-1},\qquad B=P^{-1}\left(\begin{array}{c|c} J & 0\\ \hline 0 & I_{n_a}\end{array}\right)Q^{-1},$$

exists with identity matrices I_{n_d} *and* I_{n_a} *of dimension* $n_d\times n_d$ *and* $n_a\times n_a$*, resp., with* $n_d+n_a=n$*.* $P,\ Q\in\mathbb{R}^{n\times n}$ *denote regular matrices,* $N\in\mathbb{R}^{n_a\times n_a}$ *is a nilpotent matrix and*

$$J=\begin{pmatrix} R_J & 0\\ 0 & N_J\end{pmatrix}\in\mathbb{R}^{n_d\times n_d},$$

with a regular submatrix R_J *and a nilpotent submatrix* N_J*.*

If Eq. (1) has differential index 1 and $f(t,x(t))\equiv f(t)$ then $N=0_{n_a\times n_a}$. In this case, the WKCF can be calculated with small effort with a procedure [11] which was implemented with the computer-algebra system `Maple` [4].

For the construction of the new integration scheme, the exact solutions of certain test equations are needed. With the WKCF of the matrix pair (C,B), the analytical solutions $x^{(h)}$ and $x^{(i)}$ $(i\in\mathbb{N}_0)$ of the DAE for the right-hand side $f(t,x(t))\equiv 0$ and $f(t,x(t))=\mathbb{1}t^i$, resp., with $\mathbb{1}:=(1,\ldots,1)^{\mathrm{T}}$, can be written in the following way:

$$x^{(h)}(t)=Q\left(\begin{array}{cc|c} \mathrm{e}^{-R_J t} & 0 & 0\\ 0 & \mathrm{e}^{-N_J t} & 0\\ \hline 0 & 0 & 0\end{array}\right)Q^{-1}x_0,$$

$$x^{(i)}(t)=x^{(h)}(t)$$

$$+ Q \operatorname{diag} \left(\begin{array}{c} \mathrm{e}^{-R_J t} \left[\mathrm{e}^{R_J s} \sum_{j=0}^{i} (R_J^{-1})^{j+1} \frac{(-1)^j i! s^{i-j}}{(i-j)!} \right]_{s=t_0}^{s=t} \\ \hline \mathrm{e}^{-N_J t} \sum_{j=0}^{m-1} N_J^j \left[\frac{s^{j+i+1}}{j!(j+i+1)} \right]_{s=t_0}^{s=t} \\ \hline \sum_{j=0}^{\min(i,l-1)} N^j \frac{(-1)^j i! t^{i-j}}{(i-j)!} \end{array} \right)^{\mathrm{T}} P \mathbb{1},$$

with $m = \operatorname{range} N_J$ and $l = \operatorname{range} N$.

Besides the exact integration for polynomial right-hand sides, which is similar to classical multistep methods, the exact integration of the homogeneous DAE reflects the Principle of Coherence. Hersch [8] formulated this principle as "Successive approximations should not contradict each other." The meaning of this is best shown in an example. Consider the initial-value problem $\dot{x}(t) + Bx(t) = 0$, $x(t_0) = \hat{x}(t_0)$, $t \in [t_0, t_1]$. With the increment function $\Phi(h)$ and step size $h := \frac{1}{2}(t_1 - t_0)$ we yield two numerical approximations $\hat{x}(t_1)$:

step size $2h$: $\hat{x}(t_1) = \Phi(2h)\hat{x}(t_0)$,

step size h: $\hat{x}(t_1) = \Phi(h)\hat{x}(t_0 + h) = \Phi^2(h)\hat{x}(t_0)$.

According to the principle of coherence, these approximations should be equal, leading to the condition $\Phi^2(h) = \Phi(2h)$. Therefore, $\Phi(h) = \exp(-Bh)$ holds for a coherent integration scheme.

Remark. Another interpretation of the principle of coherence is that the numerical scheme is exact for an arbitrary but fixed homogeneous test equation. This test equation can be rather complicated as long as an analytical solution can be provided.

3. Construction of an efficient integration scheme

In order to solve Eq. (1), the principle of coherence is combined with the standard multistep ansatz for $k + 1$ function evaluations, $k \in \mathbb{N}_0$,

$$\alpha_k(h) \cdot x_{n+k} + \alpha_{k-1}(h) \cdot x_{n+k-1} = h \cdot \sum_{i=0}^{k} \beta_i(h) \cdot f(t_{n+i-s}, x_{n+i-s}), \tag{3}$$

where $\alpha_{k-1}(h), \alpha_k(h) \in \mathbb{R}^{n \times n}$ and $\beta_0(h), \ldots, \beta_k(h) \in \mathbb{R}^{n \times n}$, resp., denote the coefficients of the integration scheme, $h > 0$ denotes the step size. x_v, $v \in \mathbb{N}$, denotes the numerical approximation for the exact solution $x(t_v)$ at the discrete timestep $t_v := t_0 + v \cdot h$. $s = 0$ defines an implicit and $s = 1$ an explicit integration scheme. The Principle of Coherence is reflected by the computation of the coefficients $\alpha_{k-1}(h)$ and $\alpha_k(h)$ in such a way, that the scheme integrates the homogeneous part of (1) exactly. The remaining coefficients $\beta_0(h), \ldots, \beta_k(h)$ are calculated as in standard approaches, i.e., the DAE (1) with a polynomial right-hand side $f(t, x(t)) = \mathbb{1} \cdot t^i$, where $0 \leqslant i \leqslant k$, will be integrated exactly. By using the WKCF of the matrix pair (C, B) and assuming equidistant step size $h = t_{n+1} - t_n$, this approach leads in the case of DAE (1) with differential index 1 to the following integration scheme

for $k+1$ function evaluations:

$$x_{n+k} = \underbrace{Q\left(\begin{array}{c|c} \mathrm{e}^{-Jh} & 0 \\ \hline 0 & I_{n_a} \end{array}\right) Q^{-1}}_{=-\alpha_{k-1}(h)} x_{n+k-1} + h\sum_{i=0}^{k} \underbrace{Q\left(\begin{array}{c|c} \beta_{i,d}(J,h) & 0 \\ \hline 0 & \beta_{i,a}(I_{n_a},h) \end{array}\right) P}_{=\beta_i(h)} f(t_{n+i-s}, x_{n+i-s}),$$

where $\alpha_k(h) = I_n$, $\beta_{i,d}(J,h)$ is a power series in J and h, and $\beta_{i,a}(I_{n_a},h)$ is a linear function in I_{n_a} and $1/h$ [11]. In the following, the presented integration scheme will be called MUNICH (*mu*ltistep method for the *n*umerical solution of *i*ndex 1 DAEs evaluating the principle of *c*oherence due to *H*ersch). In the case of 2 function evaluations we get

$$\beta_0(h) = -Q\left(\begin{array}{c|c} \sum_{j=0}^{\infty} \frac{(-1)^j}{(j+2)!} J^j h^j & 0 \\ \hline 0 & I_{n_a} \frac{1}{h} \end{array}\right) P,$$

$$\beta_1(h) = Q\left(\begin{array}{c|c} \sum_{j=0}^{\infty} (-1)^j \frac{j+3}{(j+2)!} J^j h^j & 0 \\ \hline 0 & I_{n_a} \frac{1}{h} \end{array}\right) P$$

for an explicit scheme, and

$$\beta_0(h) = Q\left(\begin{array}{c|c} \sum_{j=0}^{\infty} (-1)^j \frac{j+1}{(j+2)!} J^j h^j & 0 \\ \hline 0 & -I_{n_a} \frac{1}{h} \end{array}\right) P,$$

$$\beta_1(h) = Q\left(\begin{array}{c|c} \sum_{j=0}^{\infty} \frac{(-1)^j}{(j+2)!} J^j h^j & 0 \\ \hline 0 & I_{n_a} \frac{1}{h} \end{array}\right) P$$

for an implicit scheme.

Next, we analyze MUNICH with respect to its consistence, stability, convergence, and its suitability to solve stiff and highly oscillatory DAEs. It is possible to exploit some well-known definitions and theorems from classical numerical analysis.

Definition 3 (*Consistency*). The linear multistep method

$$\sum_{i=0}^{k+s} a_i x_{n+i-s} = h\sum_{i=0}^{k} b_i f(t_{n+i-s}, x_{n+i-s}) \tag{4}$$

is called *consistent of order* p, if

$$\mathscr{L}(x,t,h) := \sum_{i=0}^{k+s} a_i x(t+(i-s)h) - h\sum_{i=0}^{k} b_i \dot{x}(t+(i-s)h) = \mathcal{O}(h^{p+1})$$

holds for all sufficiently regular functions $x(t)$.

Theorem 4. *For $k > 0$ the implicit and the explicit discretization scheme MUNICH is consistent of order $k+1$. For $k = 0$ the differential part of MUNICH is consistent.*

Proof.

- The first thing to do is to identify the operator $\mathscr{L}$ and the numerical integration scheme (3) with the help of $f(t,x) = C\dot{x} + Bx$. This is in the implicit case done by

$$\begin{aligned} a_i &:= -h\beta_i(h)B && \text{for } i = 0,\dots,k-2,\\ a_i &:= \alpha_i(h) - h\beta_i(h)B && \text{for } i = k-1,k,\\ b_i &:= \beta_i(h)C && \text{for } i = 0,\dots,k, \end{aligned}$$

and in the explicit case by

$$\begin{aligned} a_i &:= -h\beta_{i+1}(h)B \quad \text{for } i = 0,\dots,k-1,\\ a_k &:= \alpha_{k-1}(h) - h\beta_k(h)B, \qquad a_{k+1} := \alpha_k(h),\\ b_i &:= \beta_i(h)C \quad \text{for } i = 0,\dots,k. \end{aligned}$$

- To prove the order of consistence k we use the following theorem (cf. [7, p. 370]) in combination with the transformations mentioned above and the fact that the integration scheme is exact for the right-hand side of the polynomial $f(t,x(t)) = \mathbb{1}t^i$ with $i = 0,\dots,k$.

Theorem. *The multistep method* (4) *is at least of consistence order p, iff*

$$\sum_{i=0}^{k+s} a_i = 0 \quad \textit{and} \quad \sum_{i=0}^{k+s} a_i (i-s)^q = q \sum_{i=0}^{k} b_i (i-s)^{q-1}$$

for $q = 1,\dots,p$.

- To verify that the order of consistence is $k+1$, it can be shown that the coherent multistep scheme with $k+1$ function evaluations is exact even for the polynomial right-hand side $f(t,x(t)) = \mathbb{1}t^{k+1}$ of degree $k+1$ (the exactness for polynomials of degree 0 to k is automatically fulfilled due to the construction of the new coherent integration scheme).
- To show that the order is not greater than $k+1$, one can use the results for the Adams–Bashforth/–Moulton schemes which are special cases of MUNICH ($C = 1$, $B = 0$).
- The special case $k = 0$ shows some difficulties. With the exact solution $x^{(1)}$ for the right-hand side of the polynomial $f(t,x(t)) = \mathbb{1}t$ of degree 1 we have

$$\mathscr{L}(x^{(1)},0,h) = Q \left(\begin{array}{c|c} \mathcal{O}(h^2) & 0 \\ \hline 0 & \mathcal{O}(h) \end{array}\right) P\mathbb{1}.$$

Here we see, that the differential part of MUNICH is consistent, but this is not true for the algebraic part. □

Theorem 5. *The implicit and explicit discretization scheme MUNICH is stable.*

Proof. The proof is straightforward and therefore omitted. □

Definition 6 (*Convergence*). The linear multistep method (4) (with the numerical solution $x_h(t)$) is called *convergent of order* p, if for the initial value problem $\{\dot{x}(t)=f(t,x(t)), x(t_0)=x_0\}$ with sufficiently differentiable f, there exists a $h_0 > 0$ such that

$$\|x(t) - x_h(t)\| \leqslant Ch^p \quad \text{for } h \leqslant h_0,$$

whenever the starting values satisfy

$$\|x(t_0 + jh) - x_h(t_0 + jh)\| \leqslant C_0 h^p \quad \text{for } h \leqslant h_0,\ j = 0, 1, \ldots, k + s - 1,$$

with C and C_0 independent of h.

Theorem 7. *For $k>0$ the implicit and the explicit discretization scheme MUNICH is convergent of order $k+1$. For $k=0$ the differential part of MUNICH is convergent.*

Proof. Similar to the theory of numerical standard schemes for the solution of DAEs, Theorem 7 follows from consistency (Theorem 4) and stability (Theorem 5). □

We now discuss the properties of MUNICH when applied to stiff ODEs. Consider the *test equation of Dahlquist*

$$\dot{x}(t) = \lambda x(t), \quad x(0) = 1,$$

and the iteration formula

$$x_{n+1} = R(h\lambda)x_n,$$

which is created by the discretization scheme applied to Dahlquist's test equation. R is called the *stability function.*

Theorem 8. *The implicit and the explicit discretization scheme MUNICH is (independent of its order) A-stable, i.e., its stability domain $S:=\{z \in \mathbb{C}: |R(z)|<1\}$ satisfies $S \subseteq \mathbb{C}^- := \{z \in \mathbb{C}: \operatorname{Re} z < 0\}$.*

Proof. The iteration formula for MUNICH applied on Dahlquist's test equation is

$$x_{n+1} = \mathrm{e}^{\lambda h} x_n,$$

so the stability function can be written as

$$R(z) = \mathrm{e}^z.$$

This shows that $|R(z)|<1$, iff $\operatorname{Re} z<0$. So the stability region for MUNICH is

$$S = \mathbb{C}^-. \qquad \square$$

In order to investigate the properties of MUNICH with respect to highly oscillatory circuits, we consider the following second-order test equation for oscillatory problems:

$$\ddot{z}(t) = -\lambda^2 z(t),$$

with initial values $z(t_0)=z_0$ and $\dot{z}(t_0)=\dot{z}_0$. Formulating this test equation as a first order ODE leads to

$$\underbrace{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}}_{=:C}\underbrace{\begin{pmatrix}\dot{z}(t)\\ \dot{y}(t)\end{pmatrix}}_{=\dot{x}(t)}+\underbrace{\begin{pmatrix}0 & -\lambda\\ \lambda & 0\end{pmatrix}}_{=:B}\underbrace{\begin{pmatrix}z(t)\\ y(t)\end{pmatrix}}_{=:x(t)}=\underbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}_{=:f(t,x(t))}.$$

Due to the construction, MUNICH solves this first-order ODE exactly, which indicates the capability of MUNICH for efficiently solving highly oscillatory DAEs.

It is clear that the efficiency of MUNICH depends on the influence of the right-hand side $f(t,x(t))$ on the solution. If the influence is 'small' as in the example above, MUNICH is much more efficient than standard schemes. If the solution of Eq. (1), however, is mostly determined by $f(t,x(t))$, then the gain in efficiency will become smaller. In the worst case ($B=0$), MUNICH will coincide with standard schemes with a somehow larger overhead. This overhead arises mostly from the computation of the (matrix) coefficients which increases with the dimension n of the system. Therefore, the appropriate application of MUNICH will be for small but highly oscillatory DAEs. The simulation of small quartz circuits yields this type of equation, an example is given in the next section.

4. Numerical experiments

The discretization scheme MUNICH is implemented in Fortran77. The implementation uses a fixed step size and fixed order scheme with order 3 (except of the first and the second integration step, which is of order 1 and 2, resp., due to the short step size history). MUNICH can be used as a purely explicit method or as a predictor–corrector method with explicit MUNICH as predictor (P) and implicit MUNICH as corrector (C) [11].

A quartz-controlled oscillator (cf. Fig. 1) was used to test the integration scheme. The modeling of the circuit is taken from [5] where a similar integration scheme for ODEs is described. The circuit consists of 4 MOSFETs (metal oxide semiconductor field effect transistors) *MPI*, *MNI*, *MNT*, *MPT*, two capacitances *CL*1, *CL*2, and the quartz, which is modeled by a resistor *RQ*, an inductance *LQ*, and two capacitances *C*0, *CQ*.

The modeling (where the MOSFETs were modeled according to level 1 of Siemens circuit simulator TITAN [6]) leads – after elimination of the trivial equations for the voltage source – to a five-dimensional system of type (1), where

$$C=\begin{pmatrix}C0+CL1+COXWL_{MPI}+\\ \quad +COXWL_{MNI} & -C0 & 0 & 0 & 0\\ -C0 & C0+CL2 & 0 & 0 & 0\\ 0 & 0 & CQ & -CQ & 0\\ 0 & 0 & -CQ & CQ & 0\\ 0 & 0 & 0 & 0 & LQ\end{pmatrix},$$

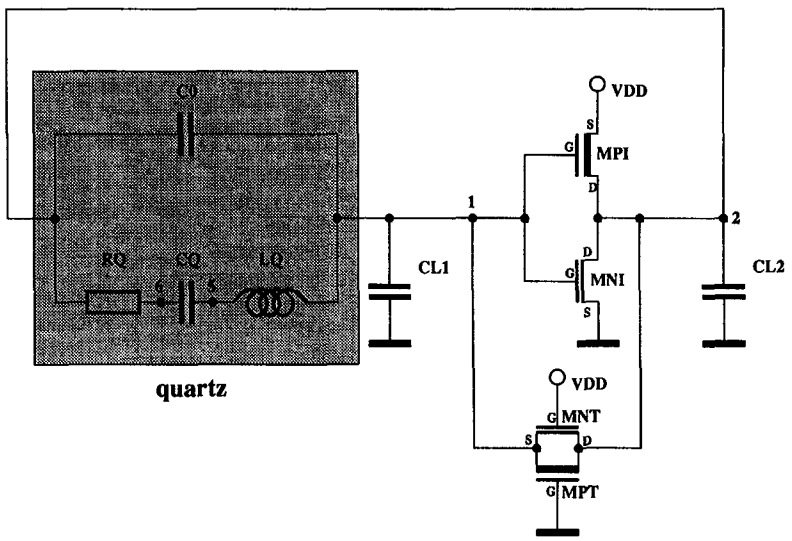

Fig. 1. Quartz-controlled oscillator model.

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & \frac{1}{RQ} & 0 & -\frac{1}{RQ} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{1}{RQ} & 0 & \frac{1}{RQ} & 0 \\ 1 & 0 & -1 & 0 & 0 \end{pmatrix},$$

$$f(t,x(t)) = \begin{pmatrix} -CDRAIN_{MPT}(t,x) + CDRAIN_{MNT}(t,x) + \\ + I_{BS_{MPT}}(t,x) + I_{BS_{MNT}}(t,x) \\ CDRAIN_{MPI}(t,x) - CDRAIN_{MNI}(t,x) + \\ + CDRAIN_{MPT}(t,x) - CDRAIN_{MNT}(t,x) + \\ + I_{BD_{MPI}}(t,x) + I_{BD_{MNI}}(t,x) + \\ + I_{BD_{MPT}}(t,x) + I_{BD_{MNT}}(t,x) \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad x(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_5(t) \\ u_6(t) \\ I_{LQ}(t) \end{pmatrix}.$$

$CDRAIN_m$ is nonlinear in x and describes the drain current of the MOSFET $m \in \{MPT, MNT, MNI, MPI\}$, the remaining functions I are nonlinear functions in x for the current sources within the MOSFET model. In this particular example, $f(t,x(t)) \equiv f(x(t))$ holds.

Looking at row 3 and 4 of matrix C, it is obvious that C does not have full rank. Therefore, the index of the system is at least 1. Differentiating the sum of row 3 and 4 and replacing row 4 yields

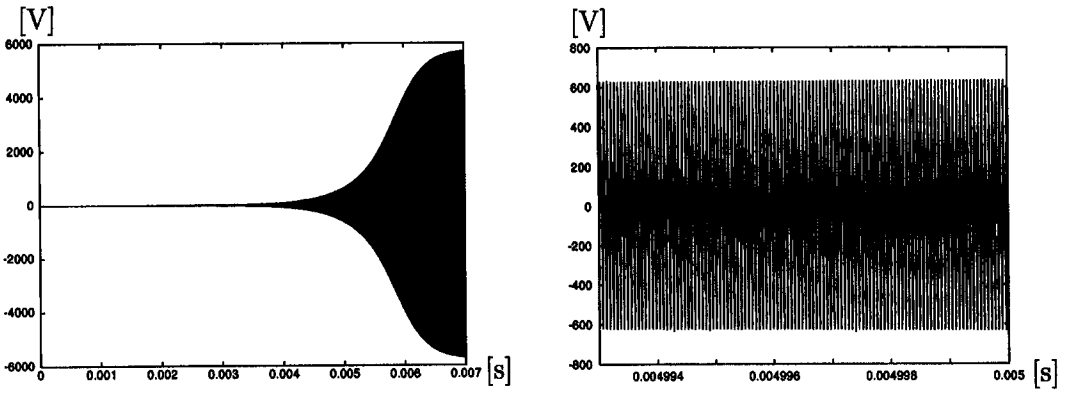

Fig. 2. Transient curve of u_5: Whole integration interval (left-hand side) and zoom-in (right-hand side).

Table 1
Simulation of the oscillator circuit: computational effort (constant step size)

Integration scheme	Expl. MUNICH	P-C-MUNICH	DASSL-3-C
Step size	1.0×10^{-9} s	2.5×10^{-9} s	1.0×10^{-10}
Integration steps	7 000 000	2 800 000	70 000 000
Function evaluations	7 000 001	7 540 768	70 000 053
CPU time (SGI Indigo^2 R4400)	431 s	350 s	10 135 s

a regular matrix and – after inverting this regular matrix – an explicit ordinary differential equation. Hence, the index of the DAE is 1.

For the consistent initial value $x(0)=(2.5, 2.5, 2.5, 2.5, 0)^T$, the oscillator was simulated in the interval [0 ms, 7 ms], which corresponds to about 115 000 oscillations. Fig. 2 shows the highly oscillating voltage progression at node 5 for the regarded time interval.

To investigate the efficiency of MUNICH, the simulation was compared with the results of DASSL-3-C. This is the well-known standard scheme DASSL [2] for DAEs, limited to maximum order 3 (variable order) and fixed step size. As MUNICH is A-stable also in the explicit case for each order of convergence, it is possible to solve the stiff circuit equations with the explicit version of MUNICH. In addition, we have investigated the predictor–corrector version P-C-MUNICH of MUNICH. The statistics of the simulations are given in Table 1.

The explicit version of MUNICH gives a sufficiently accurate solution with a step size of 1.0×10^{-9}. The step size can be increased to 2.5×10^{-9} for P-C-MUNICH, but the number of function evaluations remains nearly the same. Due to the larger number of integration steps, the explicit MUNICH requires more CPU time than the predictor–corrector scheme. In order to yield a sufficiently accurate solution it is necessary to choose the (fixed) step size for DASSL-3-C as 10^{-10}. DASSL-3-C needs about 10 times as many function evaluations as the MUNICH schemes, which is reflected in a factor of about 25 in CPU time. DASSL-3-C operates mostly at order 2 or 3. Even if one

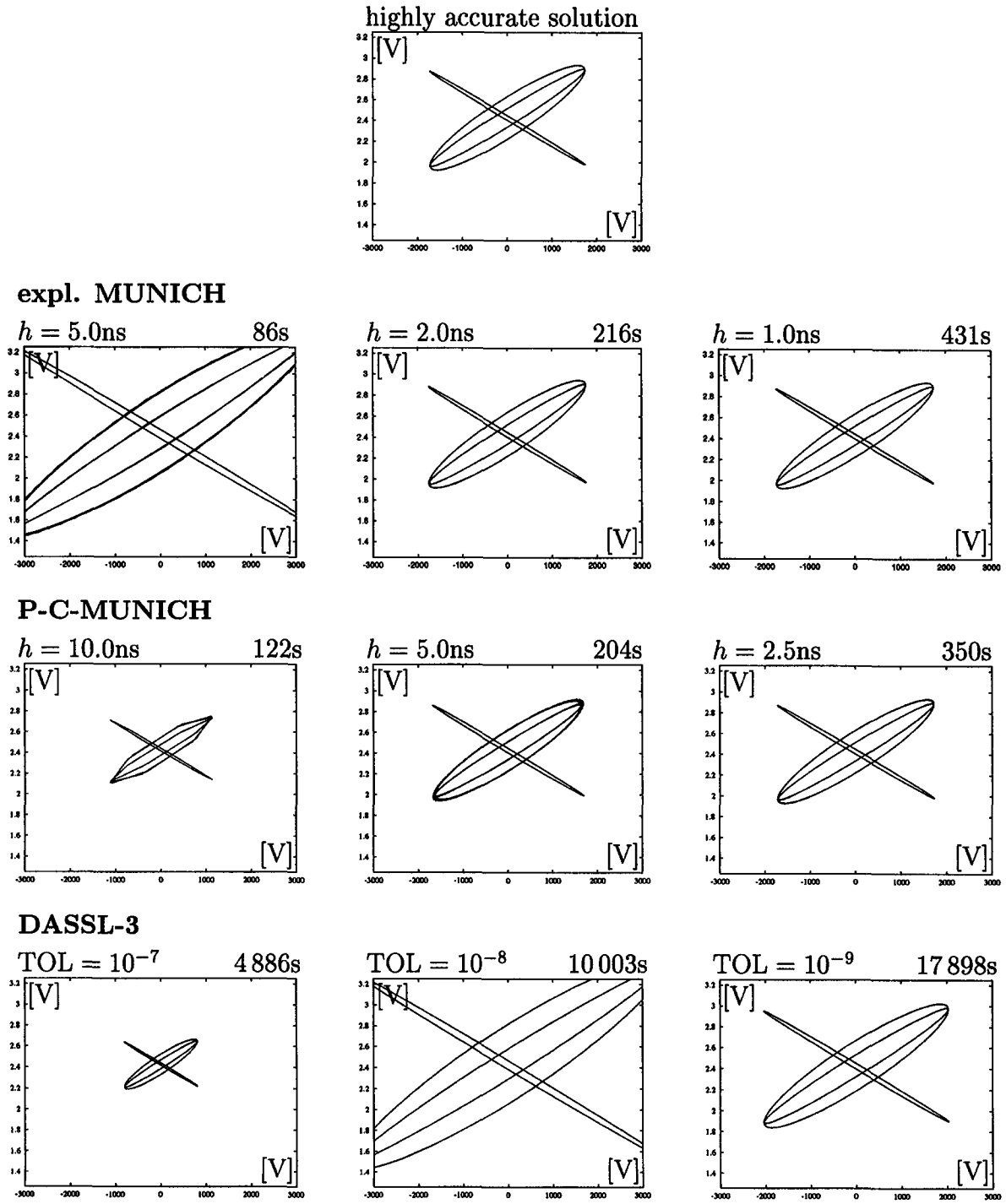

Fig. 3. Phase diagrams at $t = 5.5$ ms for various step sizes h (explicit MUNICH (top), P-C-MUNICH (middle)) and error tolerances TOL (DASSL-3 (bottom)). The CPU time necessary for the whole simulation is given for each plot.

Table 2
Simulation of the oscillator circuit: computational effort (variable step size)

Integration scheme	DASSL-3	DASSL-5
Tolerance	10^{-10}	10^{-9}
Integration steps	198 877 110	25 053 186
Function evaluations	240 327 072	42 331 566
CPU time (SGI Indigo^2 R4400)	30 601 s	4 496 s

takes the overhead for the order control in DASSL-3-C into account, MUNICH is much more efficient.

The simulation of the quartz oscillator was repeated with a variable step size version of DASSL, see Table 2. Here the error tolerances were set again in such a way that the simulation results were sufficiently accurate. If the maximum order of DASSL is increased to 5, the scheme DASSL-5 can use higher orders and larger step sizes. This proves that the restriction of the step size is not due to a failure in the step size control. Even compared with DASSL-5, the constant step size implementation MUNICH is more efficient.

Besides efficiency, reliability is an important property for integration schemes for highly oscillatory DAEs. In Fig. 3, the prescribed error tolerance (DASSL-3) and the fixed step size (MUNICH) are varied. A step size of 5 ns for the explicit MUNICH and 10 ns for the P-C-MUNICH gives too few data points for a smooth plot. With a smaller step size, these artifacts disappear.

Reducing the step size for MUNICH results in a more accurate solution, whereas strengthening the error tolerance may yield a more inaccurate (!) solution (cf. $\mathrm{TOL} = 10^{-7} \rightarrow \mathrm{TOL} = 10^{-8}$).

5. Conclusions

In this paper the numerical integration scheme MUNICH was presented. The construction of MUNICH combined a multistep ansatz with the principle of coherence due to Hersch. Due to its theoretical properties (A-stability, exact integration of the test equation for P-stability), it is able to solve differential–algebraic equations of index 1 with highly oscillatory solutions efficiently. The efficiency of MUNICH has been shown for a quartz-controlled oscillator. Compared with the standard integration scheme DASSL, the computational time was speeded up by about 80 times. Therefore, MUNICH is a useful tool for the simulation of highly oscillatory circuits since it reduces the simulation time considerably.

Acknowledgements

The authors are indebted to the Numerical Analysis group of R. Bulirsch of the TU München for helpful discussions. We also thank U. Feldmann and Q. Zheng from the Corporate Research & Development of Siemens AG, München. This work was partly supported by the Bayerische

Forschungsstiftung within the 'Bavarian Consortium for High Performance Scientific Computing – FORTWIHR' and is part of the project 'Numerical simulation of electric circuits and semiconductor devices'.

The authors acknowledge the valuable remarks of the referees which helped to improve this paper.

References

[1] G. Vanden Berghe, H. De Meyer, A modified Numerov integration method for second order periodic initial-value problems, Internat. J. Comput. Math. 32 (1990) 233–242.

[2] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical solution of initial value problems in differential-algebraic equations, North-Holland, New York, 1989.

[3] S.L. Campbell, C.D. Meyer, Generalized inverses of linear transformations, Pitman, London, 1979.

[4] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt, Maple V: language reference manual, Springer, New York, 1990.

[5] G. Denk, The simulation of oscillatory circuits: an efficient integration scheme, in: H. Neunzert (Ed.), Progress in industrial mathematics at ECMI 94, Wiley, New York, and Teubner, Stuttgart, 1996, pp. 295–300.

[6] U. Feldmann, U. Wever, Q. Zheng, R. Schultz, H. Wriedt, Algorithms for modern circuit simulation, Archiv für Elektronik und Übertragungstechnik 46 (1992) 274–285.

[7] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer, Berlin, 1993.

[8] J. Hersch, Eine Kohärenzforderung für Differenzengleichungen, in: J. Albrecht, L. Collatz (Eds.), Numerische Methoden bei Differentialgleichungen und mit funktionalanalytischen Hilfsmitteln, International Series of Numerical Mathematics 19 (1974) 121–124.

[9] P.J. van der Houwen, B.P. Sommeijer, Explicit Runge–Kutta–Nyström methods for oscillatory problems, SIAM J. Numer. Anal. 26 (1990) 414–429.

[10] W. Kampowsky, P. Rentrop, W. Schmidt, Classification and numerical simulation of electric circuits, Surveys on Mathematics for Industry 2 (1992) 23–75.

[11] Chr. Penski, Effiziente numerische Methoden zur Lösung rasch oszillierender differential-algebraischer Gleichungen vom Index 1 mit Anwendungen in der Schaltungssimulation, dissertation, TU München, Mathematisches Institut, 1995.

[12] T.E. Simos, A.D. Raptis, Numerov-type methods with minimal phase-lag for the numerical integration of the one-dimensional Schrödinger equation, Computing 45 (1990) 175–181.